The ASCI Data Models and Formats Effort

Progress Towards A Comprehensive Approach to Interoperable Scientific Data Management and Analysis

SC '98

Larry Schoof

Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy
under Contract DE-AC04-94AL85000

This is a presentation describing work in progress on the Accelerated Strategic Computing Initiative (ASCI) Data Models and Formats (DMF) effort. The work is being performed at Lawrence Livermore National Lab (LLNL), Los Alamos National Lab (LANL), and Sandia National Lab (SNL).

Outline

- Points of Contact
- Motivation
- Objectives
- Data Abstractions
- Fiber Bundle Model
- Implementation
- Status
- Future

This is the outline of the presentation.

DMF Points of Contact

- **Laboratory POCs**
  - **LLNL: Mark Miller (miller@viper.llnl.gov)**
  - **LANL: John Ambrosiano (ambro@lanl.gov)**
  - **SNL: Larry Schoof (laschoo@sandia.gov)**
- **Key collaborators**
  - **Limit Point Systems, Inc.: David Butler (dmbutler@limitpt.com)**
  - **NCSA: Mike Folk (mfolk@ncsa.uiuc.edu)**

These are the primary points of contact.

Motivation

- **Data sharability; application interoperability**
  - ASCI “ONE program, THREE laboratories” philosophy
- **Software reuse**
  - Common data format allows development of common tools
- **Leverage tri-lab experiences**
  - EXODUS (SNL), SILO (LLNL), netCDF (UCAR), PDB (LLNL), HDF (NCSA)
  - previous data abstractions were too restrictive, not robust

Why are we doing this?

To facilitate data communication between applications within (intra-lab) and among (inter-lab) the three ASCI labs, we must collaborate and leverage each other’s experiences.

A common data format allows the use of common tools for multiple application codes, rather than developing separate tools for each specific physics code. This software reuse is a common thread through the ASCI program.

We have extensive experience with storage of computer simulation data, including EXODUS and SILO, as well as lower-level data storage libraries such as netCDF, PDB, and HDF. These past efforts were successful to varying degrees, but they all suffer from one primary shortcoming: the data abstractions were too restrictive. In the current effort, we have learned from our past experiences and have devoted considerable resources to create a data abstraction that doesn’t have the limitations of past efforts.

Motivation (cont.)

- **Workshop on Interoperability of DOE Visualization Centers**
  - Hosted by DOE MICS (Dan Hitchcock); ANL, BNL, INEEL, LBNL, LLNL, LANL, NASA-Ames, ORL, PNNL, SNL, NCSA
  - 4 recommendations; 2 concerned with ASCI DMF
    - non-ASCI DOE sites participate in DMF workshops
    - national labs incorporate DMF into their visualization codes, once it is proven
- **Collaboration with Petroleum Open Standards Consortium (POSC)**
  - Establishing standards for data exchange

As an example of groups in the simulation community who are interested in data sharing, DOE (MICS office) has initiated an informal effort to encourage interoperability of visualization tools. At the first workshop held in March, the ASCI DMF project was discussed as a related effort that could result in software libraries that facilitate data sharing between applications. Half of the recommendations that came out of that workshop concerned ASCI DMF.

In a recent meeting of the Petroleum Open Standards Consortium, a resolution was passed to collaborate with the ASCI DMF effort.

Objectives

- **Sound data model with robust data abstractions**
  - **Address computational physics data types**
    - **Fields on meshes (structured, unstructured, hierarchical)**
  - **Extendible**
- **Common data format**
  - **Allows common tools (setup, analysis, visualization, etc.)**
  - **Allows data communication among tools**
- **Common application programming interface (API)**
  - **Shield applications from complexities of data model**
  - **Perform permutations (e.g., domain local <--> global mapping)**
- **High performance**
  - **Reduce ratio of data manipulation time / compute time**

What do we intend to accomplish?

1. Develop data abstractions for computational physics data types (fields on meshes); create a data model (objects and operators) that uses these abstractions.

2. Develop data schema (tables, trees, graphs, etc.) that implements the data model and is stored in a common format that facilitates the use of common tools.

3. Develop a “high-context” (using the semantics of the application domain) programming interface that performs the mapping from the application domain to the data model.

4. Tune the software for the three ASCI hardware platforms.

Low-context Data Abstractions

- **netCDF (UCAR)**
  - multi-dimensional arrays of ints, floats, etc
- **HDF4 (NCSA)**
  - Scientific Data Set: multi-dimensional arrays of ints, floats, etc
  - Vdata: collection of records of fixed-length fields (tables)
  - Vgroup: collection of related objects
  - raster: 2-dimensional raster image
  - palette: color-lookup table with 256 entries
  - annotation: text attached to objects
- **HDF5 (NCSA and ASCI labs)**
  - multi-dimensional array of record structures
  - grouping structure

"Low-context" data abstractions are those that contain little or no references to the application domain. Application domain objects may be mapped to the lower abstractions, but the context is lost in the mapping. For example, a coordinate field may be represented as an n-dimensional array, but the fact that the values in the array are positions of points in space is lost.

Because these abstractions don't contain domain-specific objects, they can represent data from many domains, and thus serve a valuable purpose. However, the context must be preserved with some other mechanism, such as attaching attributes or adopting naming conventions.

HDF5 is a clear departure from older versions of HDF in that all the previous data types are now represented with just two mechanisms: an n-dimensional array and a grouping structure. This has simplified the programming interface and may result in better performance.

High-context Data Abstractions

- **EXODUS (SNL)**
  - element blocks, connectivity arrays, coordinate arrays, element variables, nodal variables, global variables, side sets, node sets, distribution factors, QA records, information records, element maps, node maps, properties, ...
- **SILO (LLNL)**
  - quadmesh, quadvar, ucdmesh, ucdvar, pointmesh, multimat, multimesh, multivar, material, material species, zonelist, facelist, curve, variable, directory, ...
- **Data Explorer (IBM)**
  - field: a mapping from some domain to some data space
  - domain: positions and connections
  - data space: dependent variables
  - arrays and groups

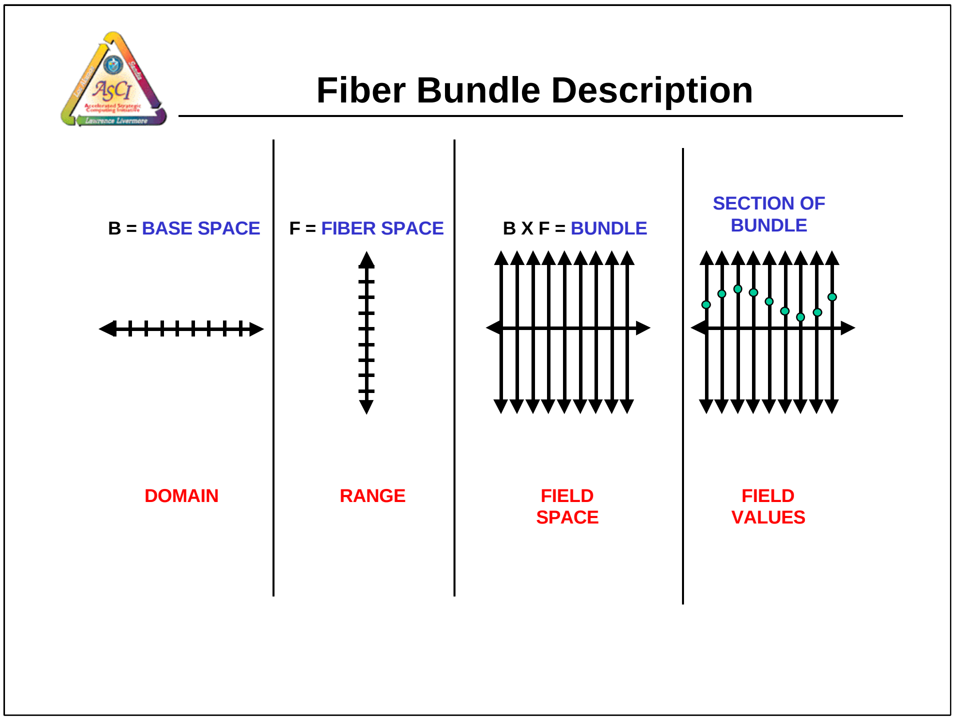
"High-context" data abstractions preserve domain-specific semantics and provide meaning to data objects. As a result, these typically are useful for a relatively small set of applications. EXODUS and SILO are examples that have proven beneficial in that they have allowed data sharing between applications within specific domains. However, their data abstractions are not robust enough to easily accommodate new types of data or even minor modifications to existing entities without adding completely new objects, resulting in a plethora of supported objects.

We have been intrigued by the IBM Data Explorer (DX) data abstraction, which is more general but provides sufficient context to be useful. Its foundation is in the mathematical field of fiber bundles.

ASCI DMF Data Abstraction

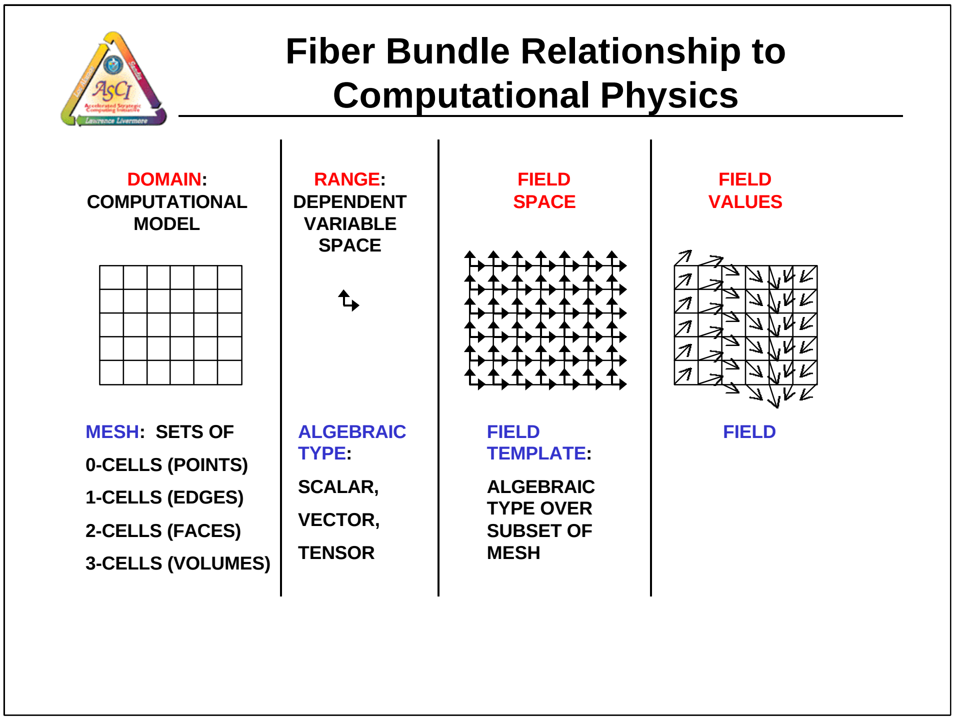
- **High-context (i.e., meshes and fields)**
- **Based on mathematical field of fiber bundles**
- **Previous examples**
  - IBM Data Explorer data model
  - LPS WHITNEY visualization system
  - NASA Super Glue
  - UCLA Conquest
  - UCSD/DEC Sequoia 2000

As a result of much investigation, we decided to base the DMF data abstraction on fiber bundles. Past successes (IBM DX and LPS Whitney) demonstrated that this had great potential to be robust enough to represent the high-context information from our computational simulation domains.



This and the next slide are intended to provide a brief description of *fiber bundles* and how they map to computational physics data.

This is a simple example of representing the function y = f(x) in *fiber bundle* terminology. The X axis (domain) is the *base space*. The Y axis (range) is the *fiber space*. The cartesian product of the X and Y axes form a *bundle* that represents the space where all possible values of X and Y exist (field space). Picking a specific value of Y, the dependent variable, for each value of X, the independent variable, forms a *section of a bundle* (the function or field values).



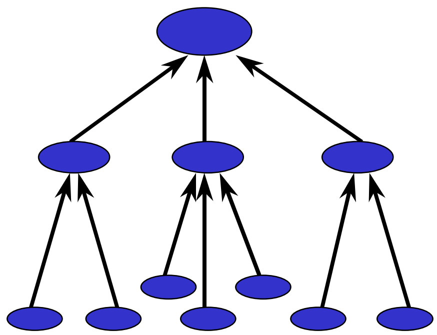
In computational physics, the domain is represented as some computational model, typically a mesh consisting of sets of points, lines, surfaces, and volumes.

The range is the dependent variable space represented as zero-order tensors (scalars), first-order tensors (vectors), and second-order tensors.

The field space is some algebraic type (scalar, vector, tensor) over a portion or all of the mesh. It represents the space in which dependent variables are valid. We refer to this as a field template.

The actual values of the dependent variables compose a field.

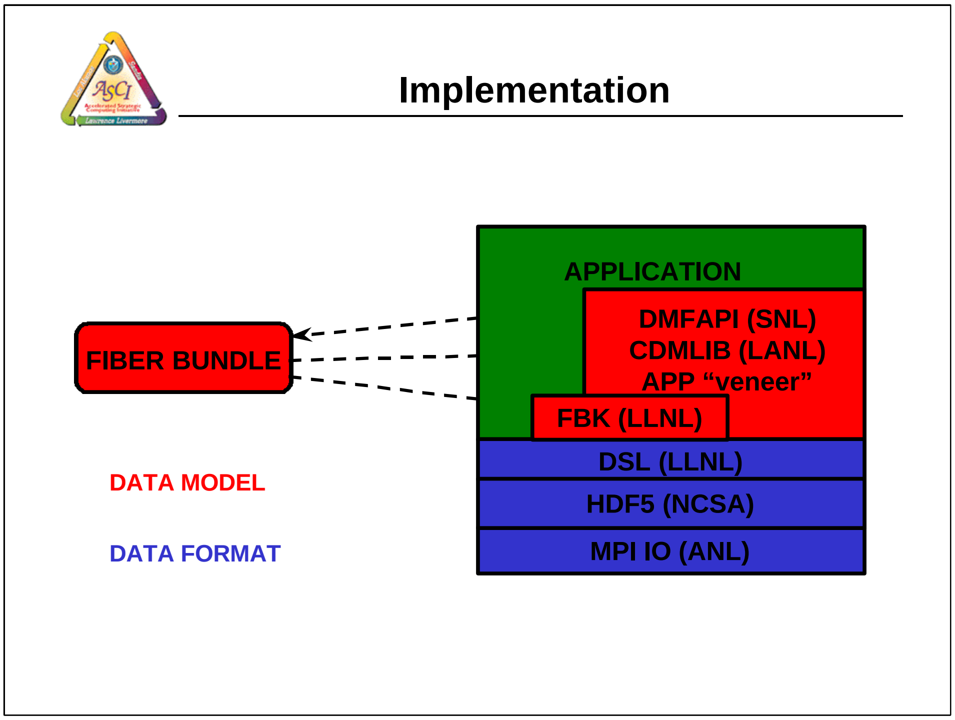
Subset Inclusion Lattice (SIL)



- Directed acyclic graph
- Nodes are sets of:
  - 0-cells (points)
  - 1-cells (edges)
  - 2-cells (faces)
  - 3-cells (volumes)
- Links are inclusion maps between subsets and supersets
- Lattice represents:
  - topology
  - assembly
  - domain decomposition
  - boundary

We chose to represent the base space (computational model) as a subset inclusion lattice. It is implemented as a directed acyclic graph in which the nodes are sets of 0-cells, 1-cells, 2-cells, and 3-cells. The links in the graph are inclusion maps that describe what cells in a superset are contained in a subset.

By assigning “roles” to these maps, the lattice is used to represent topology (for example, point lists defining volumes, point lists defining faces which define volumes, etc.); assembly (grouping sets together to form larger sets); domain decomposition (inclusion maps map process-local data to global data -- local-to-global maps); boundary (for example a list of faces that form the external surface of the mesh).



This depicts the software layers developed to implement the data model. The layers in red deal primarily with the data model; the ones in blue deal with the data format. The arrows imply that the selection of the fiber bundle data model was driven by the types of data in our application domains, and that once chosen, the data model drove the development of two of the software layers.

A brief description of each layer follows:

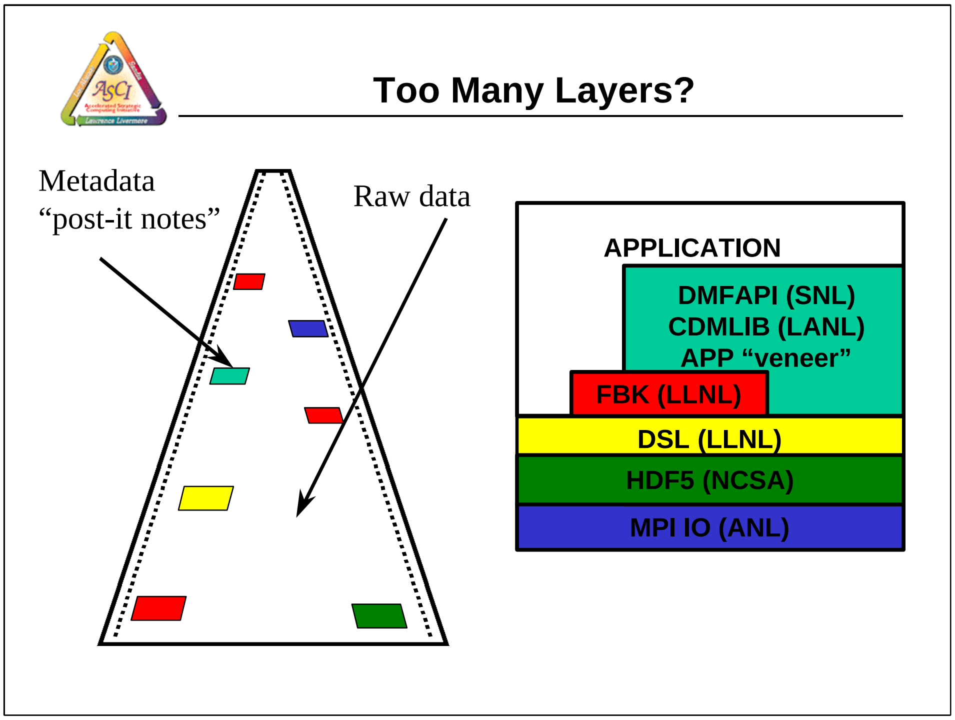
MPI IO (Message Passing Interface I/O): This is a portion of the most recent specification of MPI (MPI2) that includes parallel (collective and independent) I/O. We selected this specification as it is a likely candidate to become the standard parallel I/O interface specification for distributed memory and SMP cluster machines. As vendors create and tune custom implementations of the specification, we will benefit with little or no changes to higher layers. Our beta software is using the ROMIO implementation of MPI2 from Argonne National Labs.

HDF5 (Hierarchical Data Format 5): HDF5 is a complete rewrite of previous versions of HDF in collaboration with the ASCI labs. This layer specifies the format of the data on the storage medium.

DSL (Data Structures Layer): The primary purpose of this layer is to isolate the underlying layers that actually perform I/O from the upper layers. This facilitates switching the I/O layer if it becomes necessary.

FBK (Fiber Bundle Kernel): This implements the data model (as relations or tables) in terms of fiber bundles (i.e., fibers, bundles, etc.).

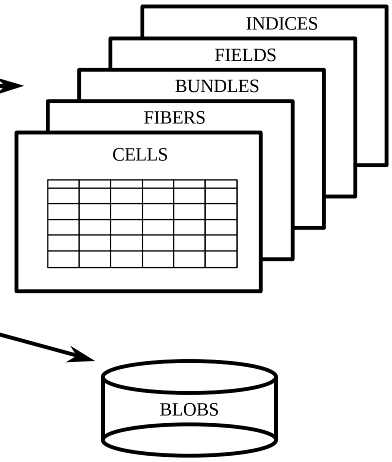
DMFAPI / CDMLIB (DMF Application Programming Interface / Common Data Model LIBrary): These are prototype “high-context” layers that provide the mapping from application domain-specific data to entities in the fiber bundle model. Additional transformations such as processor local-to-global mapping are also performed at this layer.



Some people are concerned about performance of our implementation with so many layers. Think of the raw data represented as numbers on reams of computer paper. To describe the data, metadata is attached as “post-it notes” to give the raw data some context (i.e., “This vector of data are the coordinates.” or “This array of ints is the local-to-global map for processor 0.”). The raw data is passed through each of the layers untouched, unless the client requests a transformation to be performed.

Performance Considerations

- **Light data (metadata)**
  - **memory resident**
  - **data is local (private) or global (shared) across processors**



- **Heavy data (raw data)**
  - **file resident**
  - **no transformations unless requested**

The current implementation of the Fiber Bundle Kernel uses five tables (cells, fibers, bundles, fields, indices) to store the metadata. This "lightweight" data is typically on the order of a few megabytes. The tables are memory resident and each entry in each table is private or shared according to the client's request. The raw data (typically 3-4 orders of magnitude more than the metadata) is passed through all of the layers straight to disk, unless a transformation (for example, a gather or scatter) is requested by the client.

Status

- **Completed**
  - two releases of unified data model and common data schema
  - two releases of data schema implementation (VBT 1.0)
  - alpha, beta, and full releases of parallel HDF5
  - designed and implemented prototype "high-context" API's
- **Future**
  - deliver top-to-bottom solution (beta) (FY99 Q1)
  - refine unified data model and common data schema (FY99 Q2)
  - refine "high-context" API (FY99 Q3)
  - tune for better performance (FY99 Q3)
  - deliver top-to-bottom solution ("production") (FY99 Q4)

This summarizes our accomplishments to date and our future deliverables.

Future

- **Encourage other agencies to adopt**
- **Become vehicle for interoperability (visualization, analysis, meshing, etc.)**
- **Become de facto standard for sharing data within computational physics community**

Our goal is to provide a vehicle that facilitates application interoperability in the computational physics community.